

EXHIBIT A

Google is committed to advancing racial equity for Black communities. [See how.](#)

Firebase > Docs > Guides



Log User Actions

[Send feedback](#)

Google Search uses information about the actions users take on public and personal content in an app to improve ranking for Search results and suggestions. To improve your users' experience when they search for content in your app, log user actions through the App Indexing API.

What actions should you log?

Use the following guidelines as you log user actions:

- Log the user's interactions with the app, including viewing content, creating new content, or sharing content.
- Only log actions the user takes on content directly — not background actions within the app like incoming messages or playlist syncs.
- Don't log actions for each item in a list when a user interacts with the entire list. For example, don't log view actions for each message whenever a user opens their inbox.

 **The App Indexing API has changed:** Follow the steps in our [migration guide](#) to update the Firebase App Indexing library if you're using a version older than Firebase App Indexing 10.0.0.

Build and log the actions

To build the `Action` object, define the following parameters:

- Title and URL of the content
- Appropriate `Action.Builder` for the action type

Then, log the action:

- For actions that take place over a longer duration, call both the `start()` and `end()` methods, accordingly.
- For instantaneous actions, call the `end()` method as soon as the user takes the action. For example, when a user comments on a note, log an instantaneous action with a single call of the `end()` method.

Keep personal content on device

When logging actions, keep data for personal content on device. By default, action data is uploaded to Google servers with the user's consent (learn more about [user consent](#)). Make sure you explicitly flag actions on personal content that shouldn't be uploaded by calling `setUpload(false)` on the `Metadata` object. Users can view and delete data about actions they've taken on public content from Google servers at <https://myactivity.google.com/>.

About action types: Use the correct `Action.Builder` constant for your content. For example, use the `VIEW_ACTION` constant for opening static content and the `WATCH_ACTION` constant for playing video content. See [a list of constants](#) for the `Action.Builder` class.

Regarding fragments: You structure fragments the same way the activity in the example below is structured. But, because fragments may execute many times within an activity, or there may be multiple fragments, take care to make the API call only once. Here are some guidelines to follow:

- If the activity calls the API, then don't call the API again from any fragment within the activity.
- If the activity doesn't call the API, and you want a fragment to call it instead, then make sure only one fragment calls the API, only one time.

[Start and end](#) Instantaneous

The example below uses public content.

```
Java      Kotlin+KTX
Android   Android

@Override
protected void onStart() {
    super.onStart();
    // If you're logging an action on content that hasn't been added to the index yet,
    // add it first.
    // See <a href="https://firebase.google.com/docs/app-indexing/android/personal-content#up">
    FirebaseUserActions.getInstance().start(getRecipeViewAction());
}

@Override
protected void onStop() {
    FirebaseUserActions.getInstance().end(getRecipeViewAction());
    super.onStop();
}
```

MainActivity.java [↗](#)

[Next: Test Your Implementation](#)

Was this page helpful?



[Send feedback](#)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see the [Google Developers Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2021-02-01 UTC.

Documentation

[OVERVIEW](#)[GUIDES](#)[REFERENCE](#)[SAMPLES](#)[LIBRARIES](#)[SEND FEEDBACK](#)[Get Started](#)[Analytics](#)[DEVELOP](#)[Cloud Messaging](#)[Cloud Functions](#)[Authentication](#)[Realtime Database](#)[Storage](#)[Hosting](#)[Test Lab](#)[Performance Monitoring](#)[Crash Reporting](#)[GROW](#)[Remote Config](#)[App Indexing](#)[Introduction](#)[iOS](#)[Android](#)[Get Started](#)[Enable Public Content Indexing](#)[Enable Personal Content Indexing](#)[Log User Actions](#)[Test Your Implementation](#)[Enhance Search Performance](#)[Migrate to the Latest API](#)[Dynamic Links](#)[Invites](#)[AdWords](#)[EARN](#)[AdMob](#)

Log User Actions

Google Search uses information about the actions users take on public and personal content in an app to improve ranking for Search results and suggestions. To improve your users' experience when they search for content in your app, log user actions through the App Indexing API.

What actions should you log?

Use the following guidelines as you log user actions:

- Log the user's interactions with the app, including viewing content, creating new content, or sharing content.
- Only log actions the user takes on content directly — not background actions within the app like incoming messages or playlist syncs.
- Don't log actions for each item in a list when a user interacts with the entire list. For example, don't log view actions for each message whenever a user opens their inbox.

The App Indexing API has changed: Follow the steps in our [migration guide](#) to update the Firebase App Indexing library if you're using a version older than Firebase App Indexing 10.0.0.

Build and log the actions

To build the `Action` object, define the following parameters:

- Title and URL of the content
- Appropriate `Action.Builder` for the action type

Then, log the action:

- For actions that take place over a longer duration (e.g., viewing a recipe), call both the `start()` and `end()` methods, accordingly. For example, you would log separate calls for viewing a recipe (start) and then closing the recipe (end).
- For instantaneous actions, call the `end()` method as soon as the user takes the action. For example, when a user comments on a note, log an instantaneous action with a single call of the `end()` method.

About action types: Use the correct `Action.Builder` constant for your content. For example, use the `VIEW_ACTION` constant for opening static content and the `WATCH_ACTION` constant for playing video content. See [a list of constants](#) for the `Action.Builder` class.

Regarding fragments: You structure fragments the same way the activity in the example below is structured. But, because fragments may execute many times within an activity, or there may be multiple fragments, take care to make the API call only once. Here are some guidelines to follow:

Contents

What actions should you log?
Build and log the actions

When you log actions, remember to:

Make sure you've added content to the index: If you're logging an action on content that hasn't been added to the index yet, add it first. For more information about adding content to the index, see [Update the index](#).

Keep action data for personal content on device: By default, action data is uploaded to Google servers with the user's consent (learn more about [user consent](#)). Make sure you explicitly flag actions on personal content that shouldn't be uploaded by calling `setUpload(false)` on the `Metadata` object. Users can view and delete data about actions they've taken on public content from Google servers at <https://myactivity.google.com/>.

- If the activity calls the API, then don't call the API again from any fragment within the activity.
- If the activity doesn't call the API, and you want a fragment to call it instead, then make sure only one fragment calls the API, only one time.

START AND END

INSTANTANEOUS

The example below uses public content.

```
@Override
public void onStart() {
    super.onStart();

    // If you're logging an action on content that hasn't been added to the index yet,
    // add it first.
    // See https://firebase.google.com/docs/app-indexing/android/personal-content#update-the-index


    FirebaseUserActions.getInstance().start(getRecipeViewAction());
}

@Override
public void onStop() {
    FirebaseUserActions.getInstance().end(getRecipeViewAction());
    super.onStop();
}
```

NEXT: TEST YOUR IMPLEMENTATION

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated May 17, 2017.

Cloud Functions	▼
Authentication	▼
Realtime Database	▼
Storage	▼
Hosting	▼
Test Lab	▼
Performance Monitoring 	▼
Crash Reporting	▼
GROW	
Remote Config	▼
App Indexing	▲
Introduction	
▶ iOS	
▼ Android	
Get Started	
Enable Public Content Indexing	
Enable Personal Content Indexing	
Log User Actions	
Test Your Implementation	
Enhance Search Performance	
Migrate to the Latest API	
Developer Links	▼

Learn

Guides
Reference
Samples
Libraries
GitHub

Stay Connected

Blog
Dev Summit
Facebook
Google+
Twitter
YouTube

Support

Contact Support
Stack Overflow
Google Group
Release Notes
Known Issues
Legacy Documentation



Android
Chrome
Firebase
Google Cloud Platform
All Products